Assign a Rover To a Mission and Move Mission from RDY to OUT	Choose correct rover type (ER/MR/PR), ensure rover is available, and match mission requirements. link rover to mission, remove rover from available list.	**3**	Yousef
Mission OUT to EXEC	When a mission reaches it target	**2**	Yousef
Mission EXEC to BACK	When execution ends, move mission to BACK list	**2**	Yousef
Mission BACK to DONE	When mission returns, mark DONE and handle its rover	**2**	Omar
Rover to Avail/Checkup (Handle different cases when a rover needs maintenance)	When rover reaches the base station, check if maintenance is required; send to checkup list. Otherwise, send to available list	**3**	Omar
Rover from Checkup to Available	When checkup period ends, rover returns to the appropriate available rovers list.	**2**	Abdallah
Abort Request handling AbortRequest::Operate( ) calls MarsStation::AbortMission(missionID)	MarsStation::AbortMission(missionID) should - if NM in RDY list, implement and call RDY_NM::Abort - if NM in OUT list, implement and call OUT_missions::Abort	**4**	Abdallah
AutoAbort poalr missions	if PM waits > double its duration	**1**	Abdallah
Comprehensive Simulation function (full station logic)	Unlike Phase1.2, must implement ALL algorithmic logic described in Phase 2; no randomness	**4**	Mohammad
Interface modes	no code repetition; output must match program interface section	**1**	Omar
UI class functions (phase1.2 feature)	Print details on the output screen	**-2 if still not impelmented**	
Generate output file and the required statsitics	Must follow required format: mission stats, rover stats, performance measures, etc.	**3**	Mohammad
Six comprehensive test cases	Must cover various scenarios, edge cases, and large loads (50+, 75+, etc.)	**1**	Omar